



AEROSPACE INFORMATION REPORT

AIR6276™

Issued 2026-03

Use of Models and Tools for Aircraft Systems Development - A Strategy for Development Assurance Aspects with Examples

RATIONALE

The purpose of this SAE Aerospace Information Report (AIR) is to explore the use of models and tools within the system development process to provide added confidence that the product will perform its intended function. Through a few examples that demonstrate key principles, this document aims to show that the issues of model and tool usage at the system-level differ from the software and complex electronic hardware item-level issues and require different considerations. The examples presented in this document show that the use of models and tools at the system-level results in a higher fidelity system definition than could otherwise be achieved, and that the risks associated with the use of these models and tools are understood and contained.

TABLE OF CONTENTS

1.	SCOPE.....	3
1.1	Purpose.....	3
1.2	Background.....	4
2.	REFERENCES.....	4
2.1	Applicable Documents.....	4
2.1.1	SAE Publications.....	4
2.1.2	EASA Publications.....	5
2.1.3	EUROCAE Publications.....	5
2.1.4	FAA Publications.....	5
2.1.5	RTCA Publications.....	5
2.2	Definitions.....	6
2.3	Abbreviations and Acronyms.....	7
3.	USE OF MODEL AND TOOL CONSIDERATIONS IN SYSTEM DEVELOPMENT.....	7
3.1	Overview.....	7
3.2	System Model and Tool Role in the Development Process.....	8
3.2.1	System Models.....	8
3.2.2	System Tools.....	10
3.2.3	Models and Tools in System Development in Comparison to Item Development.....	10
3.2.4	Role of System Models in Requirements Validation and Implementation Verification.....	10
3.2.5	Role of System Models in Modification Impact Analysis.....	11
3.3	Model and Tool Planning.....	11
3.4	Management of Error Risks Related to System Models and Tools.....	12
3.4.1	Contribution of Models and Tools in System Development.....	12
3.4.2	Recognizing Potential Errors.....	13
3.4.3	Determining the Consequence of Errors.....	14
3.4.4	Deciding on Mitigations.....	14
3.4.5	Examples of Model and Tool-Related Error and Mitigation Strategies.....	19

SAE Executive Standards Committee Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2026 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, or used for text and data mining, AI training, or similar technologies, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 1-877-606-7323 (U.S. and Canada only)
Tel: 1-724-776-4970 (outside U.S. and Canada)
Fax: 724-776-0790
Email: CustomerService@sae.org
http://www.sae.org

SAE WEB ADDRESS:

For more information on this standard, visit
<https://www.sae.org/standards/content/AIR6276/>

4.	NOTES	22
4.1	Contribution Acknowledgement	22
4.2	Revision Indicator.....	22
APPENDIX A	SYSTEM DEVELOPMENT EXAMPLE: A CONTROL SYSTEM EVALUATION	23
APPENDIX B	SYSTEM DEVELOPMENT EXAMPLE: AVIONICS NETWORK EVALUATION	31
Figure 1	Use of a model or tool in system development process framework	8
Table 1	Potential mitigation strategies by error type.....	20

1. SCOPE

1.1 Purpose

The increased use of models in the development of complex aircraft and systems provides great opportunities and benefits, but also introduces some additional risks. The purpose of this document is to clarify ways to identify, prioritize, and mitigate risks associated with the use of models and tools in aircraft and system development. This document introduces considerations for the usage of [models](#) and [tools](#) in aircraft and [system](#) development activities that are defined in ARP4754/ED-79 (at latest revision). Throughout this document, a model refers to an abstract representation of a given set of aspects of a system/function/item, and a tool refers to an application or commercial product that is used for aircraft or system development activities such as developing, managing, and executing models, managing requirements [validation](#) and implementation [verification](#) activities and associated data, and automation of complex development tasks. The characteristics of models and tools and how they are applied differently for system-level development compared to item-level development are discussed. A discussion of several types of potential [errors](#) that can arise during the application of models and tools to system development is included as well. This is followed by proposed activities for managing the risks associated with the application of models and tools to system development. These suggestions start by identifying the contribution of models and tools to the decision-making process, and then understanding the impact of potential model and tool errors, leading to a discussion of potential process mitigations and methods for increasing the confidence in the application of models and tools to system development.

NOTE: Unless otherwise stated, in the context of this document, the term “risk” is used to mean risks associated with potential errors that can be introduced in model and tool development and usage.

System development activities that may employ models and tools include but are not limited to:

- Developing the system functions, architecture, and design
- Validating system functions and their requirements under normal and unusual conditions (extreme cases, tolerance stack-ups, [failure](#) insertion, etc.), which can help define additional requirements for features to constrain adverse operation
- Developing test procedures to be used in verifying the implemented system when it becomes available
- Analyzing the performance of a system against requirements over a range of inputs and conditions or combinations thereof to supplement the system verification
- Confirming the intended behavior of a system design using a validated model

This document applies to models that are used for requirements analysis, design analysis, or to better understand and define requirements. Models that are used as the requirements are not discussed in this document, as in this case the ARP4754/ED-79 (at latest revision) Requirements Capture section and other associated sections provide the pertinent objectives and guidelines.

This document does not describe the software or [hardware](#) development process or the direct verification of such [items](#). It is assumed that these software or hardware processes, which may involve use of models, are handled by their respective guidance (e.g., DO-178C/ED-12C or DO-254/ED-80). This use of models is not described in this document, despite that it might involve system development processes in combination with software or hardware development processes.

There is already a large and growing list of models and tools being applied in the development of complex systems, including computer aided design (CAD) tools for installation analysis, models and tools supporting system safety analysis, architectural models supporting various analyses, and tools used to simulate and analyze system performance or behavior.

Two examples are presented to illustrate where models and tools are used to facilitate a set of typical system design decisions. The examples also identify model and tool errors, describe possible mitigation methods used to increase confidence in the use of models and tools in system development, and minimize the effect of errors that may be introduced by using the models and tools. The two examples are:

- An aircraft-level control law design targeted for software implementation on a digital computer, where a model is used for analysis of the control gains (referred to as the Roll Control System and described in [Appendix A](#))
- An avionics network architecture where the tool analyzes the interfaces between the elements and identifies issues in the architecture (referred to as the Network Analysis Tool and described in [Appendix B](#))

The examples herein are not intended to be an exhaustive treatment of all considerations, but rather to demonstrate a set of principles and techniques that are generally applicable to the use of models and tools in system design. The Roll Control System and the Network Analysis Tool examples show how confidence in the models and tools can be established.

1.2 Background

Given the growth in complexity of aerospace systems, it has become necessary to use modeling, [simulation](#), and computer-based engineering tools to support system development activities. The extensive use of models and tools has raised some concerns regarding the potential errors that the use of models and tools could introduce or fail to detect in aircraft or system design. This document offers examples of model and tool usage in system development to illustrate mitigation techniques that might be used to minimize the possibility of model and tool errors being introduced into the system design.

The issues surrounding model and tool usage at the aircraft and system-level are substantively different from the issues around model and tool usage at the item-level. Also, some models and tools may span the aircraft, system, and item-level development activities and related guidelines. It is important that the transition from the system-level to item-level is identified and managed in ways that are appropriate for each level. This document shows both the differences and the transitions through the examples presented.

2. REFERENCES

2.1 Applicable Documents

The following publications form a part of this document to the extent specified herein. The latest issue of SAE publications shall apply. The applicable issue of other publications shall be the issue in effect on the date of the purchase order. In the event of conflict between the text of this document and references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 1-877-606-7323 (U.S. and Canada only) or 1-724-776-4970 (outside U.S. and Canada), www.sae.org.

ARP4754B Guidelines for Development of Civil Aircraft and Systems

ARP4761A Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment

2.1.2 EASA Publications

Available from European Union Aviation Safety Agency, Konrad-Adenauer-Ufer 3, D-50668 Cologne, Germany, Tel: +49 221 8999 000, www.easa.europa.eu.

AMC 25.1309 Equipment, Systems and Installations EASA Acceptable Means of Compliance

2.1.3 EUROCAE Publications

Available from EUROCAE, The MINT Building, 2 Avenue François Mitterrand, 93210 Saint-Denis, France, Tel: +33 1 49 46 19 65, www.eurocae.net.

ED-12C Software Considerations in Airborne Systems and Equipment Certification

ED-79B Guidelines for Development of Civil Aircraft and Systems

ED-80 Design Assurance Guidance for Airborne Electronic Hardware

ED-135 Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment

ED-215 Software Tool Qualification Considerations

2.1.4 FAA Publications

Available from Federal Aviation Administration, 800 Independence Avenue SW, Washington, DC 20591, Tel: 866-835-5322, www.faa.gov.

AC 20-152A Development Assurance for Airborne Electronic Hardware

2.1.5 RTCA Publications

Available from RTCA Inc., 1150 18th Street, NW, Suite 910, Washington, DC 20036, Tel: 202-833-9339, www.rtca.org.

DO-178C Software Considerations in Airborne Systems and Equipment Certification

DO-254 Design Assurance Guidance for Airborne Electronic Hardware

DO-330 Software Tool Qualification Considerations

2.2 Definitions

This section provides definitions for terms used in this document. Citations are provided to other industry sources when usage in this document is consistent with the definition in the referenced source material.

COMPLEX ITEM: An item implemented with complex electronic hardware (refer to AC 20-152A) or software.

ERROR: An omitted or incorrect action by a manufacturer, crew member, or maintenance person, or a mistake in requirements, design, or implementation (derived from AMC 25.1309). (Reprinted from ARP4754B/ED-79B.)

NOTE: In the context of this document, error refers to a mistake in the development and application of system models or tools in the system development process that can potentially result in requirements or design error.

FAILURE: An occurrence which affects the operation of an aircraft, system, equipment, item, or piece-part such that it can no longer function as intended (this includes both loss of function and malfunction). Note: Errors may cause Failures, but are not considered to be Failures. (Reprinted from ARP4754B/ED-79B.)

HARDWARE: The physical realization of systems, equipment, or items. May refer to these objects individually or collectively. (Reprinted from ARP4754B/ED-79B.)

ITEM: A defined and bounded set of either (one or more) hardware elements or (one or more) software elements which are treated as a single entity for analytical purposes. (Reprinted from ARP4754B/ED-79B.)

MODEL: An abstract representation of a given set of aspects of a system/function/item that is used for its analysis, implementation, simulation, or code generation and that has unambiguous, well-defined syntax and semantics. (Reprinted from ARP4754B/ED-79B.)

SIMULATION: Exercising the behavior of the model to verify or to analyze the design, or to aid in requirements capture.

SYSTEM: A defined combination of subsystems, equipment, or items that perform one or more specific functions. (Reprinted from ARP4754B/ED-79B.)

TOOL: An application or commercial product that is used for aircraft or system development activities such as developing, managing, and executing models, managing requirements validation and implementation verification activities and associated data, producing reports, performing automated checks for data consistency, and automation of complex development tasks.

VALIDATION: Specific types of validation include:

- **MODEL VALIDATION:** The determination that the model definition is correlated with real system observations or independent results with the intent to increase confidence that the model output is reasonable and accurate.
- **TOOL VALIDATION:** The determination that the tool output is reasonable and accurate.
- **REQUIREMENTS VALIDATION:** The determination that the requirements for a product are correct and complete. (Reprinted from ARP4754B/ED-79B definition of "VALIDATION.")

VERIFICATION: The evaluation of an implementation of requirements to determine that they have been met. (Reprinted from ARP4754B/ED-79B.)

2.3 Abbreviations and Acronyms

ADS	Air Data System
AMC	Acceptable Means of Compliance (EASA)
ARP	Aerospace Recommended Practice (SAE)
CAD	Computer-Aided Design
EASA	European Union Aviation Safety Agency
EUROCAE	European Organization for Civil Aviation Equipment
FAA	Federal Aviation Administration
HILS	Hardware-in-the-Loop Simulation
ICD	Interface Control Document
IOM	Input/Output Module
IRS	Inertial Reference System
RTCA	RTCA, Inc.
SILS	Software-in-the-Loop Simulation
WG	EUROCAE Working Group

3. USE OF MODEL AND TOOL CONSIDERATIONS IN SYSTEM DEVELOPMENT

3.1 Overview

The topic of model and tool assessment or qualification is not explicitly addressed within either ARP4754B/ED-79B, ARP4761A/ED-135, or their predecessors. Therefore, it is occasionally tempting for users to try to apply the guidance found in the item domain guidelines (e.g., DO-178C/ED-12C, DO-254/ED-80, DO-330/ED-215) for tool qualification. The domain for [complex items](#) is concerned with “reducing, eliminating or automating” some part of the development and verification processes. However, models and tools at the system-level are used primarily to define and improve “WHAT” needs to be created. This document focuses on the assessment and evaluation of system models and tools, especially with respect to analyzing system requirements and design. It is also recognized that models could be used to facilitate system verification, such as analyzing performance over a wide range of inputs correlated with a limited set of tests of the as-built system. A description of the types of model or tool errors will follow and a discussion of the management of system model or tool related risks will conclude this document. [Figure 1](#) outlines the framework of this document which provides guidelines for planning the model or tool purpose and scope, identifying potential error sources, and selecting the necessary mitigations.

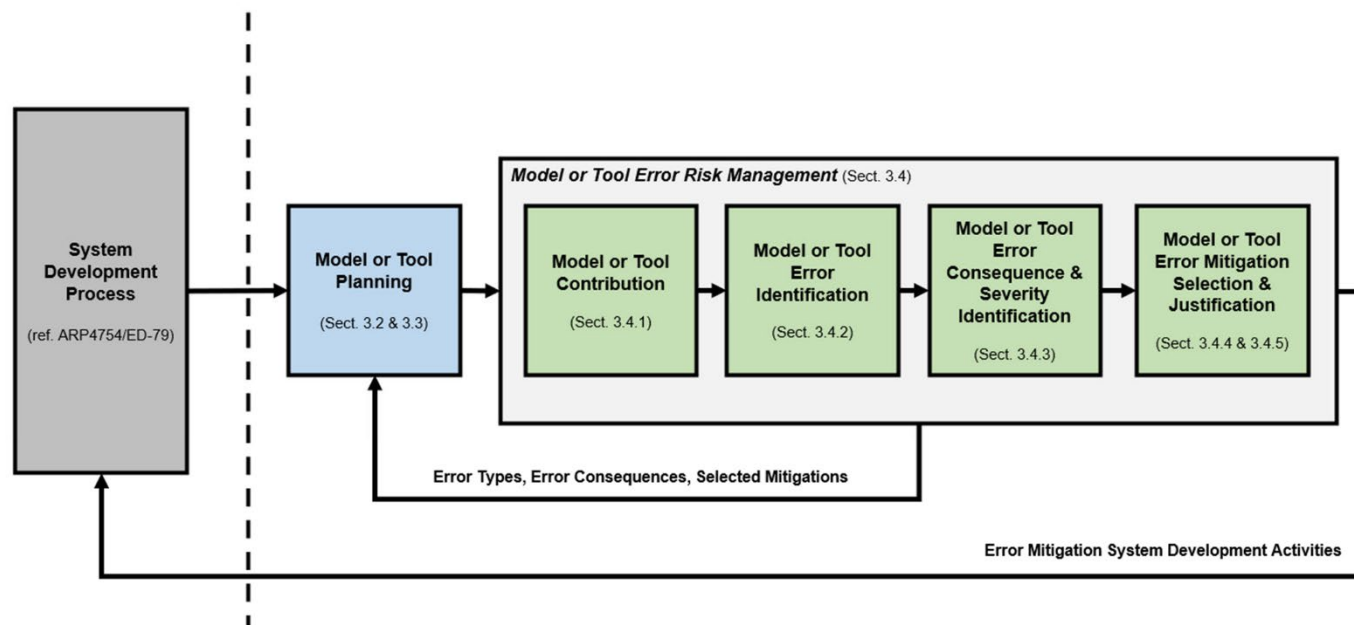


Figure 1 - Use of a model or tool in system development process framework

Systems can be developed with or without models, although some types of models have been used for decades and with increasing sophistication. Such models enhance the developers' understanding of the system, providing a means to evaluate a system's function and performance during the system concept and design phases. Models can be used to explore the design space and evaluate a range of conditions or system failures and iteratively refine the design to develop a mature system that meets the users' needs.

Note that the modeling paradigm also enables effective collaboration among design specialists; multiple experts can contribute to item or subsystem models that contain an executable encapsulation of their particular knowledge. These models can be connected together in the model of the system, such that the contributions from many experts are virtually integrated in a natural way.

In the remainder of this document, the term "model" is used to encompass system models. The term "system model" as used in this document is not addressing the use cases of either portraying requirements or code generation.

3.2 System Model and Tool Role in the Development Process

3.2.1 System Models

To move toward a common understanding of the role of system models first requires a common understanding of what is meant by a system model. The system model framework as presented herein describes the view used for this document and is not intended to be a definitive definition.

A model representing a given set of aspects of a particular system is referred to as a "system model." These system models include specific items and properties, but not necessarily all, of the system under development and the external environment as appropriate. The behavior of systems comes from the interactions of their items in addition to the individual behaviors of each item. Therefore, the interactions between the items need to be carefully considered when predicting and defining all aspects of system behavior.

Systems may also exhibit behaviors that are not desired. This behavior may be caused by failures or errors but may also come about as an unanticipated consequence of how the items interact. This emergent behavior, especially when it is unintended behavior, is one of the key challenges in engineering a system.

System functions, when implemented in a particular architecture, also have attributes that characterize the whole system. Examples of these are reliability, maintainability, functional performance, safety, security, and noise. System models play an important role in predicting and understanding multiple system attributes and their interactions early during development.

System models can be used to represent a system architecture or design and contribute to the capturing of system requirements expressed as text, block diagrams, tables, plots, or more recently, executable or dynamic models.

Principal benefits of modeling and analysis in system development include:

- Providing design-decision data earlier than it would otherwise be available, enabling the design to evolve earlier in the development cycle, thus greatly reducing costs
- Providing assessment over a greater span of conditions while making more data available by:
 - Facilitating assessment of a broader spectrum of test cases than might be practical with hardware-in-the-loop ground or flight testing, supporting greater variability and control of factors such as:
 - Flight conditions
 - Input stimuli
 - Operational scenarios, including 'abuse' cases
 - Turbulence/disturbance injection
 - Allowing assessment of the impact of equipment tolerances on overall system performance, which is impractical to do with physical prototypes
- Providing access to system internals, which can be difficult or even impossible to observe with physical prototypes
- Allowing early assessment of failure propagation across the system(s)
- Capturing system architecture with defined syntax and semantics helping to fully defining the system and communication across the stakeholders

Older system development methods had limited means to fully understand and predict the overall behavior of a system and its attributes prior to its actual implementation. Modern technology has enabled the creation of many tools that provide a platform to facilitate more comprehensive modeling of systems than traditional methods in order to achieve an understanding and prediction of system attributes and behavior. Early methods often depended on the creation of a system prototype to have a sufficient understanding of the system under development. Growing complexity, increasing software-intensive functionality, and developmental cost pressures have shifted modern system development away from dependence on physical prototyping and toward increased use of computer-based system models and tools.

3.2.2 System Tools

System models are used to gain a better understanding of systems' behavior and attributes by utilizing an abstraction of the intended system. Tools, on the other hand, are used to host or evaluate the system models. Some uses of the tools include, but are not limited to:

- Model development, i.e., providing graphical tools and rudimentary elements that can be used to model a functional or physical system
- Model execution, i.e. solving the mathematical equations the model comprises
- Efficient analysis of a large number of inputs and boundary conditions for the system
- Test or analysis data post-processing to present the results in more readable and comprehensible forms that help draw meaningful conclusions
- Model configuration management

Developers often use a system model to assist the development of a system, and typically use a tool to view and manipulate that system model. For this reason, it is not always easy to make a clear distinction between a tool and the model represented in the tool, as the two are closely related in a system development context.

The rationale and limitations for using a specific tool should be understood, including the mathematical simplifications that the tool may use in executing the models.

3.2.3 Models and Tools in System Development in Comparison to Item Development

System models are used in the decision-making process to guide the design of a system that successfully implements its intended function and to provide predictions of its emergent behavior and the attributes that it may have. Tools can be used to host these models and provide simulation capabilities. They do not directly implement the design of the items comprising the system. The outcome of system-level tool usage is an enhanced set of inputs into the decision-making process that defines the requirements for the system. Such decisions include what items will make up the system, what the architectural context of those items is, and what item functional requirements are needed for the system to perform its intended functions. These decisions necessarily guide item-level development as requirements are flowed down from the system-level to the item-level.

These system models and tools therefore bring the sense of adding something to the process, rather than, as is the concern in the domain of complex items, reducing, eliminating, or automating some part of the process. As such, the use of the system models and tools can enhance the design and validation process by providing early support and validation of design decisions, and may be used, with appropriate checks and balances put in place to minimize the possible effects that tool-related errors could have on the system. A framework to address these checks and balances is introduced in [3.4](#).

In the situation where there is a hand-off of system models to software or complex electronic hardware processes where the models are consumed directly into the design, the impact of the system modeling approach should be considered at the item-level. Here the potential model error goes into the product directly, where it did not in the system-only modeling approach, so the consequence has increased.

Use of system models that are consumed by the complex item design process directly, e.g., as requirements or for the generation of embedded code, is beyond the scope of this document.

3.2.4 Role of System Models in Requirements Validation and Implementation Verification

System models can be used to validate requirements of a proposed design or to verify that the implemented design meets their respective requirements, subject to understanding the limitations of the model and the tool used to capture the model. Such uses of models and tools may confirm the appropriateness of a system's functions or may involve analysis to show that a modeled design can meet performance requirements. The models may also be used to establish the expected results for the lower-level "as-built" implementation.

Models can be used to evaluate the system performance over a larger set of conditions than may be feasible or safe to do on the actual system, including unusual conditions (e.g., extreme cases, tolerance stack-ups, failure insertion). Such activities can further confirm the appropriateness of the system's functions, support requirement validation, and help capture additional requirements for features to constrain adverse operation.

Models can be especially valuable for identifying design sensitivities that may need closer attention in subsequent testing. Verification test cases can be conceived and dry-run on models before running them on an actual system to be better prepared for lab or aircraft testing. In addition, the models may be used to assess the coverage the test cases can achieve on the actual system so it can be determined what remains untested and the coverage gaps can be filled in. As a system design is iterated, a correlation between the model and an actual system can naturally emerge. If well established, such correlations may be used to build a case for redistributing the verification activities performed by test to simulation and other analytical methods based on mitigation techniques described in this document.

3.2.5 Role of System Models in Modification Impact Analysis

System models can help understand the impact of changes on a system by first implementing them in the model to see if they change the system as expected or cause any unexpected impacts not otherwise easily foreseen. If a model is used for Modification Impact Analysis, the configuration control of the model should be carefully planned (see [3.4.4.1](#)).

3.3 Model and Tool Planning

A model and tool planning document can be very useful to capture the strategies for the model or tool development and for addressing any associated risks. The planning document will help in documenting the following aspects of the model or tool application:

- Scope and boundaries of the models or tools
- Specific behavior or characteristics of the system with the required accuracy or fidelity that will be modeled, its intended use across the product development life cycle, and the intended benefit that the model or tool is expected to provide
- Organization structure and the respective roles and responsibilities in the development process
- Description of planned mitigation strategies identified from the risk assessment (see [3.4](#)) which may include but are not limited to:
 - Peer Reviews that will be conducted
 - Any structured activity that will be conducted to correlate the model with the real system or behavior
 - Modeling standards that will be used
 - Configuration control process and mechanisms that will be used
- Any intended use of the model or tool for showing compliance to regulatory requirements
- Major phases of the model development process and their intended outputs

Based on the risk management results, unacceptable error consequences may cause the system development team to reconsider and update the purpose and use of the model or tool in the development process or plan additional activities including prototyping or system testing.

NOTE: If a model or tool is used to meet ARP4754 objectives, then the associated processes and artifacts should be included with or within the Development Assurance planning elements per ARP4754/ED-79 (at latest revision).

3.4 Management of Error Risks Related to System Models and Tools

During the conceptual phase, system definition is more abstract in nature, which may preclude system models from providing right or wrong answers with certainty. However, as the models or tools mature and design decisions are made based on the data obtained from the models, an assessment and management of the risk associated with each decision influenced by the use of models and tools is necessary. Each of the contributors and potential mitigations should be considered as part of a larger managed-risk argument. Potential mitigations should be commensurate with the level of risk.

System model and tool errors can be identified, assessed, and addressed using standard risk evaluation methods. Managing the risks associated with the use of system models and tools fundamentally involves assessing the errors based on their potential impact and then applying appropriate mitigation to increase confidence in their usage. Specifically, the risk evaluation process may be performed by asking a set of questions, which will be discussed in the following subsections.

1. What is the expected contribution of the model result/outcome to system development activities?
2. What are the potential model and tool errors that may occur and their causes?
3. What is the consequence of these errors based on the outcome of the system development activities?
4. What mitigations could sufficiently increase the confidence that these errors do not negatively impact system development activities?

The following subsections discuss the strategies to address risk associated with the use of models and tools in the system development process. If model or tool results are directly used for compliance demonstrations against specific certification regulations, then the planned mitigation strategies from this AIR should be coordinated with the certification agencies.

NOTE: Unless otherwise stated, in the context of this document, the term “risk” is used to mean risks associated with potential errors that can be introduced in model or tool development and usage.

3.4.1 Contribution of Models and Tools in System Development

The impact a system model has on the design is dependent on how the system model is used (with or without other design activities) and for what purpose (the model's function). The model outcomes may vary between playing a significant role and providing only a starting point in an early design iteration. For example, models may play an important role in:

- Predicting and understanding the system characteristics early during development
- Providing design decision or requirements validation data earlier than it would be otherwise available
- Providing assessment over a greater span of conditions for validation and verification purposes while making more data available
- Showing compliance to regulatory requirements where testing on aircraft may not be feasible or safe
- Facilitating effective collaboration as well as building a holistic view of the system

System models may provide insight and understanding of the solution with respect to system attributes, thereby playing a large role in the overall understanding of the system and its behavior. This may have a significant impact on developing potential solutions, as system models explore the problem domain and provide a systematic trade of attributes across these potential system solutions.

System models may also provide information to help define or allocate requirements for the system, directly impacting design decisions that shape and constrain system functionality and characteristics. They can enhance the likelihood of successfully achieving the overall system functions while also providing evidence to support those design decisions.

3.4.2 Recognizing Potential Errors

System models and tools are used to understand complex problems. Their application can include multiple views, process iterations, and steps. This is part of the larger context of system analysis and system architecting. System architecting and analysis at times can be more of an art than a science given the high levels of uncertainty and ambiguity. System models and tools play an important role in reducing uncertainty and ambiguity but rarely eliminate them. Errors can be introduced given this process complexity.

System model output error is often a matter of degree rather than of absolute absence or presence. The best that can be achieved is an approximate model that may be representative of a real system or its operating environment due to factors such as the complexity of item interactions and emergent behavior in systems. It is not possible or necessary for a model to be one hundred percent accurate to provide useful insight into a system under development.

A system tool's usage is acceptable when its outcomes are sufficiently accurate to support the applicable system development activities.

As part of the risk management process, the potential error sources and their consequences for a particular usage of a model or tool should be recognized so appropriate mitigations against those potential errors can be planned. Sources of error that may impact the validity of a system model and tool output can be very different from errors related to model and tool usage for complex items. It is important to note that an error source may have one or many causes. To increase confidence that the likelihood of an error is minimized, these potential causes must be understood.

The remainder of this section will give some examples of common sources of model and tool errors which may lead to an error in system design. This is not intended to be an exhaustive or mutually exclusive list of error sources, or their causes, that may be introduced by the use of models and tools.

3.4.2.1 Development Process Errors

System model and tool development can include multiple views, large development teams, and several iterations. Lack of or insufficient processes and standards used to develop the model or tool can introduce errors. These process errors may result in incorrect integration of model components or incorrect or inaccurate representation of the system being modeled. Examples include, but are not limited to, models out of sync with the specified requirements or implemented design when a structured process is not used in their development, and different components of the model not connected correctly.

3.4.2.2 Configuration Management Errors

Although configuration management is part of the Development Process, it can be a significant source of error and is therefore highlighted separately here. Examples of configuration management errors include, but are not limited to, inconsistent versions of models or tools, tool incompatibility with the model, inconsistent management of tool configuration settings, inadequate management of multiple analytical models for multi-domain trade studies, and changes implemented without proper documentation or communication with teams. Configuration management errors may lead to the other error types and are therefore essential to address.

3.4.2.3 Representation Errors

Models are representations of certain aspects of a system and its environment that are of interest to the designer. Errors can exist in the form of misrepresentations of these system aspects and its environment, for which there can be varied reasons. For example, if there is a misperception about the system of interest on the part of the model creator, modeler, or tool developer, the model or tool created may not accurately represent the system being developed. There may also be limitations due to inadvertently capturing only key mechanisms and not others that may have an impact on system function. Representation errors may also include errors that result from the adaptation of a previously existing model (e.g., copy-and-paste error).

3.4.2.4 Computational Errors

Mathematical equations are used to depict physical behavior. Many numerical methods are used by the models or tools that solve these mathematical equations. These numerical methods can lead to errors that may impact the validity of the answer. Examples include, but are not limited to, an iterative method in the tool with an insufficient step size/sample time, precision, or convergence criteria that may lead to computational error or incorrect representation of a continuous system.

3.4.2.5 Errors Resulting from Inapplicable or Improper Usage

A model or tool may be valid under some circumstances but not valid for others. For example, a model or tool that uses Bernoulli's equation may be a valid approximation for a gas at lower velocities, but should not be used for high Mach number as it will result in invalid conclusions. Another example is using a functional model, which can be valid for understanding and defining some aspects of a system, but may not be valid for others. This could happen when the inclusion of behavior from a physical implementation, e.g., mechanical non-linearity or dissipated heat, influences the behavior of interest and would be missing from a functional model. If the inaccuracy or limitations of the model or tool are not taken into account during the use of the model, tool, or interpretation of their results, errors can be introduced in the system design.

3.4.2.6 Setup Errors

Errors can be introduced when misinterpreting the user interface or feeding incorrect inputs (e.g., boundary conditions) to the model or tool during the setup process. Setup errors can be caused by an incorrect or inconsistent understanding of the model or tool, incorrect tool parameter settings, incompatibility of hardware for running a tool, or the lack of proper documentation of the user interface. Consistency in model structure, use of a modeling standard, or use of a model or tool user guide makes it easier for model users to find the information they need in the model or tool and gives them guidance on how to use the model or tool.

3.4.2.7 Results Interpretation Errors

The results of a system model or tool may be misinterpreted and be a source of error. The results may be misinterpreted because of insufficient understanding of the model and its output. Examples include, but are not limited to, applying wrong units, not fully considering the accuracy of model output variables, and misinterpreting the meaning of model output variables (such as assuming an incorrect sensor location).

3.4.3 Determining the Consequence of Errors

Design-decision errors may occur with or without contribution from model or tool errors. However, the more a model or tool contributes to development activities (e.g., Requirements Validation, Implementation Verification), the greater the potential risk that a model or tool error may negatively impact the design or confidence in the design substantiating data. Not all model or tool errors pose the same risk of affecting the system or aircraft. Therefore, the context (e.g., functional or safety) of a design-decision error and associated model or tool errors will aid in determining the potential consequences.

The risk assessment will evaluate the consequence(s) of model or tool-related errors on design decisions and relevant development activities. The assessment should include common and foreseeable worst-case consequences that could affect either a system function or the ability to achieve a system characteristic. Examples of these consequences may include, but are not limited to, erroneous requirements or unintended system behaviors.

NOTE: The effect of a potential error on the system or aircraft may be already known (or partially known) from various sources, including the system safety assessment activities, which may be used in the risk assessment.

3.4.4 Deciding on Mitigations

Managing model and tool error-related risk involves assessing the potential errors and understanding the consequences of these errors on the product being developed. The level of confidence that the potential risk has been addressed is determined by the effectiveness of mitigation(s) that are applied. A mitigation, as used in this document, is any activity that increases the confidence that errors related to model or tool usage in system development have been addressed. Some mitigations can include existing planned system development activities. Mitigations may be preventive in nature (i.e., reducing the likelihood that an error occurs at all) or may detect errors once they have occurred, with the intent to mitigate the potential consequence. Mitigations for prevention are preferred over mitigations for detection.

Each type of error can require different mitigations to be in place to increase confidence that all potential causes of error have been addressed. Confidence in addressing the potential causes of error should also be commensurate with the associated risk of those errors (i.e., lower confidence in addressing potential causes of error will equate to higher risk with decisions made based on the model or tool results). Differences in model and tool purpose, model type, and tool selection may result in multiple mitigations being appropriate across different applications. As such, there is not a one-size-fits-all approach to sufficiently increase confidence that all potential causes of errors have been addressed.

Applicable mitigations may change depending on the phase or architecture level where the system model or tool is being used. For example, as the project transitions to the Implementation Verification phase, a higher dependency on testing and lower dependency on system models and tools may decrease the risk associated with models and tools.

The following subsections describe some, but not all, common mitigation techniques that are used to minimize model and tool errors. The need for developing and deploying specific training should be evaluated for all mitigation techniques being applied.

3.4.4.1 Model and Tool Configuration Management

System-level models and their associated tools may be created for a single use to explore an aspect of the system and then be archived. However, it is more likely they will be used and revisited multiple times across a development program and across similar systems. During these multiple uses, the models and tools will typically undergo their own life cycle of refinement and iteration. Therefore, configuration management of the models and tools becomes critical to mitigate errors that can potentially be introduced because of an obsolete or incorrect version being used.

To mitigate these errors, a configuration management approach, such as described in ARP4754/ED-79 (at latest revision), can be used. This approach may include, but is not limited to, establishing the following:

- Naming convention for models and tools to be configured
- Baseline management
- Problem reporting when errors are identified
- Change control (access control), i.e., protection against unauthorized change
- Change control tracking, i.e., what changed with respect to the previous baseline
- Configuration index, capturing names and version of models and tools
- Correlation between models, requirements, and design baselines
- Archival and recovery methods such that model and tool results can be reproduced if needed

These elements can be tailored based on the impact a system model or tool will make on design decisions and associated severity. It is recommended to establish a configuration management process for the use of models and tools.

3.4.4.2 Modeling Standards

Use of an established or approved framework to develop the models reduces development and interpretation errors. The definition and use of a modeling standard during the development phase helps address these error types. A typical modeling standard may include, but is not limited to, the following:

- Modeling language or tools used
- Modeling style conventions
 - Notations used and their definitions
 - Model elements/library definitions and allowable elements
 - Model hierarchy
 - Naming conventions for the elements in the model and any attribute associated with the model (e.g., file names, input/output names, variable names)
 - Formatting recommendations

- Model complexity guidelines or restrictions
- Constraints or limitations on tools or model library elements usage
- Methods of documenting assumptions used in the development of the models

The modeling standard can be tool or project specific.

3.4.4.3 Peer Review

Peer reviews can be used to ensure the correct usage of models and tools for the development of systems. Peer reviews may range from an individual peer using a checklist to review the model directly, to a more structured approach such as an audit panel setting. The following lists some, but not all, aspects that should be considered for such reviews:

- Peer reviews may be used to provide a range of checks across various stages of the system development process:
 - A concept review may be conducted before a specific model (and its specific version and type) or tool is selected and before modeling is initiated. This could identify any assumptions that may be required in the early stages to pursue modeling.
 - A preliminary review may be conducted once the model or tool is configured with initial conditions for check-out runs in order to ensure the correct boundary conditions and input formats have been established. This will also review assumptions as appropriate.
 - An implementation review may be conducted when the model or tool is ready to start running data generation, for example, to ensure assumptions are acceptable and there are no modeling misrepresentations.
- To guide the process of peer reviews, checklists may be developed and made available to address common sources of errors or error types.
- The number of reviewers may be selected based on the consequence of the model or tool output to the system design (see [3.4.3](#)).

3.4.4.4 Built-in Checks

Built-in checks can be implemented within the models or tools to automatically highlight issues with the use of the models and tools. Commercial tools already provide some capability for built-in checks, and more can be developed internally for the specific application. Checks could include, but are not limited to, input errors (e.g., out-of-bound input, units that are inconsistent with how the tool is going to use the input), out-of-bound results, and erroneous calculations.

3.4.4.5 Model and Tool Validation

Model and tool outcomes can be correlated with real system observations or independent results as system development progresses. Confidence in the model and tool increases with each iteration and correlation to real system attributes and behavior.

The history of models and tools that have been successfully used in the past may substantially reduce the associated risks of their future usage, provided that the limitations of the models and tools are well understood. Understanding this history may increase the confidence in model and tool usage and aid in the risk assessment.

Model correlation at physical test points can serve to establish model confidence to support validation or verification over an expanded set of test points. Some corner conditions may not be possible to correlate to test data, in which case extrapolation may be used to reduce risk.

Examples of model and tool validation methods include, but are not limited to:

- Hardware-in-the-Loop Simulation (HILS)
- Software-in-the-Loop Simulation (SILS)
- Testing (e.g., laboratory or aircraft) with correlation analysis
- Review of past model application and analysis results
- Comparison between tool outputs with separately generated (e.g., hand-calculated) results

Correlation analysis includes evaluating gaps between model results and real data. This evaluation may use various metrics such as standard deviation, root mean square error, area-under-the-curve difference percentage, etc. The gap acceptability should be commensurate with the intended use of the model.

When model and tool validation is used as a mitigation, the following aspects should be considered in defining the effort:

- Insufficient validation coverage: A history of successful use can increase confidence in the accuracy and precision of a model, the repeatability of results obtained from it, and the degree to which its use predicts real-world parameters of interest. However, the validation analysis results may likely be affected by many factors including but not limited to: system boundary conditions and system sensitivities or robustness to internal and external sources. When validation from past history does not cover the new application of the model, the model validation activity is insufficient. A well-documented model validation activity can address this type of error.
- Validation using an inappropriate real system (e.g., insufficient data collection accuracy, material and design changes not reflected on the test bench): If the model is validated with the data obtained from real hardware that is not truly representative of the system, the model validation activity may not be effective. In this case, although the model is validated, the validation result may not provide true accuracy.
- Changes implemented in models and tools: Changes may invalidate the correlation between the model and the real hardware data. Any change to the model or tool should be properly evaluated for its impact on the successful history of its usage.
- Extrapolation: There may be risk in using an extrapolation of a model or tool for a corner-condition which cannot be explicitly covered by the validation exercise. Justification for, and any known limitations of, this extrapolation should be documented.

NOTE: Application of this mitigation technique is expected when the implementation verification results used for verification substantiation are generated using system models and tools.

3.4.4.6 Downstream Implementation Verification

The aircraft or system-level implementation verification, per the guidelines in ARP4754/ED-79 (at latest revision), can provide another opportunity to validate the requirements or design decisions that were made using models and tools. For example, if performance of a controller is specified using a model, the verification methods (e.g., test, analysis, inspection) against the system-level requirements can confirm whether the performance is acceptable or not. The verification process may mitigate almost all error types described in [3.4.2](#), provided the verification is independent of the model used in the development phase. If verification is conducted against the same requirements that are generated with the support of the model, then errors due to modeling may remain hidden. Downstream verification may also validate the assumptions and simplifications used to develop the model, addressing the modeling representation errors.

Downstream implementation verification, although a viable mitigation, is not a very cost-effective one. Fixing a design issue during the verification phase is more expensive than discovering and fixing the problem in the requirements development phase. For verification to be effective for model and tool error risk mitigation, the verification activities should include testing against the system-level requirements that capture the intended functions and their performance.

3.4.4.7 Model and Tool User Guides

Model and tool user guides provide information needed by the user to ensure that the model and tools are run properly and within the constraints of the model design and tool limitations.

A typical user guide may include, but is not limited to, the following:

- Model usage
 - Model execution settings, including underlying computational methods
 - Working file structure
 - Initialization and start-up procedures, including any scripts required to be run
 - Limitations and expected accuracy
- Input definition, including:
 - Required inputs for the model to run properly
 - Optional user configurable overrides, and instructions on how to perform the override
 - Meaning of each input, and what it will be used for in the model
 - Format for entering the inputs
 - Valid ranges for the inputs
- Output definition, including:
 - Output format
 - Expected output ranges
 - Guidance for interpretation of results
- Error management
 - Definition of potential errors
 - Guidance for addressing errors
- Information on how to write scripts, including:
 - Scripting language
 - Constraints and limitations
 - Instructions for running a script
 - Configuration management process for scripts

A model user guide may be tool and project specific.

3.4.5 Examples of Model and Tool-Related Error and Mitigation Strategies

This section aligns mitigation strategies defined in [3.4.4](#) with each of the model and tool-related errors defined in [3.4.2](#). [Table 1](#) lists each of the error types along with potential mitigations and associated justification and limitations for how they might address the error types. Mitigations can be one of two types, i.e., one that minimizes incorrect results being produced by the model, and the other that improves identification of incorrect results. The potential mitigations are listed in order of their likely availability throughout the development cycle of the project. The mitigation strategies selected for a particular project may include, but are not limited to, those listed in [Table 1](#). It is up to the individual project team to determine which mitigations are most applicable for their specific circumstances. The justification and limitation column may help determine the appropriateness of mitigations for a specific project.

Table 1 - Potential mitigation strategies by error type

Error Types (see 3.4.2)	Potential Mitigations (see 3.4.4)	Mitigation Justification & Limitations
Development Process Errors	Modeling Standards	Modeling Standards can reduce the potential for introduction of errors throughout the development of the model by driving consistency.
	Peer Review (Concept)	Peer review held before development can ensure adequate processes and standards, etc., are in place and will be used to develop the model or tool.
	Model and Tool Validation	The validation activity can indicate when the model or tool results are not adequately representative of the real world. NOTE: The validation activity can't distinguish that a development process error was the cause for a discrepancy.
	Configuration Control	Configuration Control can protect against unauthorized changes and use of incorrect or obsolete versions during model or tool development.
Configuration Management Errors	User Guide	If multiple versions are available, the User Guide can inform which version is appropriate.
	Peer Review (Preliminary)	Peer Review throughout development can evaluate the details of how the system has been modeled and identify any unacceptable assumptions or representations.
	Configuration Management	Configuration Control can prevent the use of an obsolete or incorrect model version being used.
Representation Errors	Peer Review (Concept)	Early peer review can ensure the correct representation of characteristics for the system analyzed using models or tools.
	Peer Review (Preliminary)	Peer Review throughout model or tool development can evaluate the details of how the system has been represented and identify any unacceptable assumptions or representations.
	Peer Review (Implementation)	Peer Review of model or tool results can aid in identification of unacceptable representation of the system of interest.
	Model and Tool Validation	The validation activity can indicate when the model or tool results are not adequately representative of the real world. NOTE: The validation activity can't distinguish that a model representation error was the cause for a discrepancy.
	Downstream Implementation Verification	Implementation Verification can catch a number of errors including errors introduced into the design by a representation error. NOTE: Verification can't distinguish that a model representation error was the cause for a discrepancy.
	User Guide	User Guide can inform the user about the scope and limitations of the model or tool and its applicability.
Computational Errors	Peer Review (Concept)	Early peer review can ensure the correct computation of characteristics for the system to be modeled.
	Peer Review (Preliminary)	Peer Review throughout development can evaluate the details of how the system has been modeled and identify any unacceptable calculations.
	Peer Review (Implementation)	Peer Review of model or tool results can aid in identification of unacceptable calculations of characteristics relevant to the system of interest.
	User Guide	User guide can help inform or warn the user of the approximation or computational limits of the model or tool and how to identify and manage errors if they occur.

Error Types (see 3.4.2)	Potential Mitigations (see 3.4.4)	Mitigation Justification & Limitations
Computational Errors (cont.)	Built-in Checks	A built-in check can flag out-of-bounds or unrealistic results from the model.
	Model and Tool Validation	If the model or tool is validated and used within its intended validated bounds, then it can minimize the risk of computational error. NOTE: The validation activity can't distinguish that a computational error was the cause for a discrepancy.
	Downstream Implementation Verification	Implementation Verification can catch a number of errors including errors introduced into the design due to computational errors. NOTE: Verification can't distinguish that a computational error was the cause for a discrepancy.
Errors Resulting from Inapplicable or Improper Usage	User Guide	User guide can help identify applicability of available models or tools. User guide can document any restrictions or limitations of use of a model or tool (e.g., where a model has been used and where it has not yet been used).
	Peer Review (Implementation)	Early peer review can ensure the correct usage of the model or tool type.
	Model and Tool Validation	The validation activity can indicate whether the model or tool results are adequately representative of the real world. NOTE: The validation activity can't distinguish that inapplicable or improper model or tool type was the cause for a discrepancy.
	Downstream Implementation Verification	Implementation Verification can catch a number of errors including errors introduced into the design by the use of inapplicable/improper model type. NOTE: Verification can't distinguish that inapplicable or improper model or tool type was the cause for a discrepancy.
Setup Errors	User Guide	User Guide can ensure sufficient understanding of the model, tool, and their setup of initial conditions, etc., necessary for correct usage.
Results Interpretation Errors	User Guide	User Guide can ensure sufficient understanding of the model or tool, including its inherent inaccuracies, for correct interpretation of results. Also, adequate training can be considered for the application of the model or tool, especially for large and complex models or tools.

4. NOTES

4.1 Contribution Acknowledgement

The leadership of the S-18 and WG-63 Committees would like to thank the actively contributing committee members, and their sponsoring companies, for the time, effort, and expense expended during the years of development of this document. Without the experience, cooperation, and dedication of these people, development of this document would not have been possible.

4.2 Revision Indicator

A change bar (|) located in the left margin is for the convenience of the user in locating areas where technical revisions, not editorial changes, have been made to the previous issue of this document. An (R) symbol to the left of the document title indicates a complete revision of the document, including technical revisions. Change bars and (R) are not used in original publications, nor in documents that contain editorial changes only.

PREPARED BY SAE S-18 AIRCRAFT AND SYS DEV AND SAFETY ASSESSMENT COMMITTEE

APPENDIX A - SYSTEM DEVELOPMENT EXAMPLE: A CONTROL SYSTEM EVALUATION

A.1 INTRODUCTION

For this example, an Aileron Motion Controller (hereafter referred to as “aileron controller”) was developed for the Flight Controls System roll axis (Roll Control System) on the hypothetical S18 aircraft. A commercially available tool was used to develop an analysis model of the Roll Control System, including the plant models (aircraft and actuator), controller model, and pilot interface. This analysis model was primarily produced to evaluate aileron controller gain selection during the system development process (see [Section A.2](#)).

This example presents the selected mitigation strategies relevant to the model development and its application in the system development process relevant to its risk assessment (see [Section A.3](#)).

The example presented here is a basic aileron controller incorporating roll rate feedback. In order to focus on the model's impact to the system development process, the control law and Flight Controls System architecture and associated requirements have been simplified. The simplified aileron controller model is shown in [Figure A1](#).

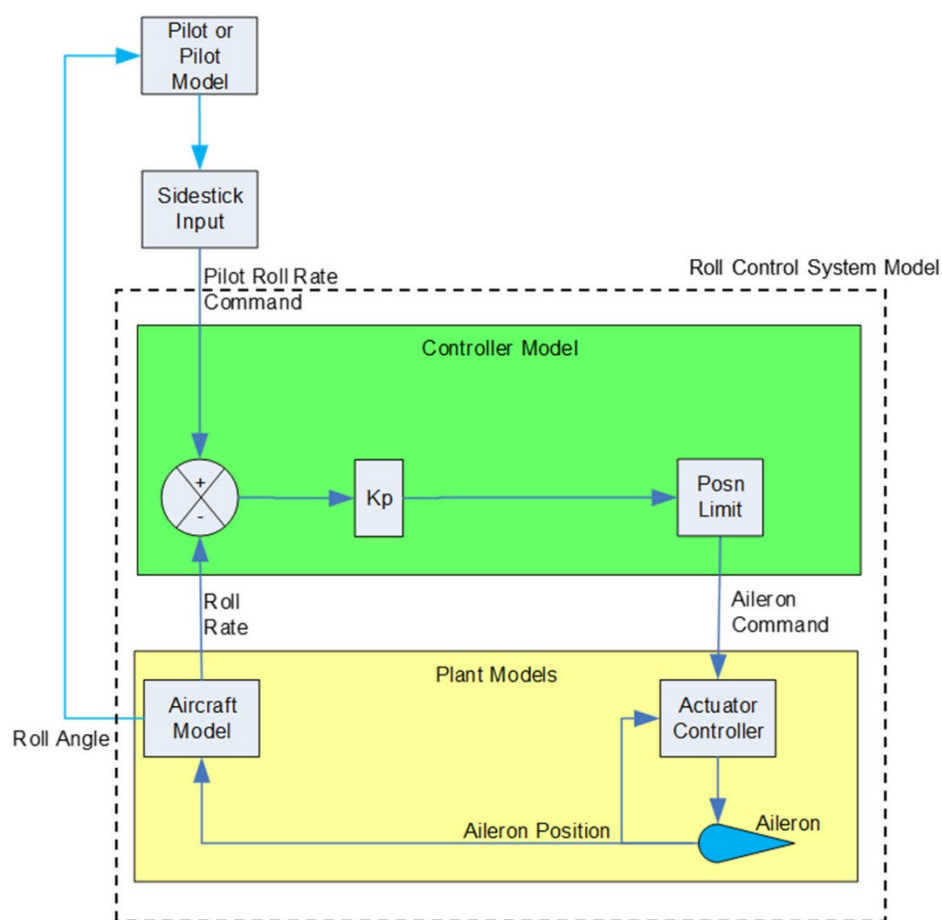


Figure A1 - Aileron motion controller model block diagram

A.2 ROLL CONTROL SYSTEM MODEL ROLE IN DEVELOPMENT PROCESS

This section of the example is intended to show how a model can be used for system analysis, system requirements capture, and validation. The model of the aileron roll axis control included both the plant model (aircraft and actuator) and the aileron controller model. This model combination was used at several points during the process of developing the Roll Control System. It should be noted that the loss of or erroneous function of the Roll Control System, as identified by the Safety Assessment process, is likely to result in effects classified as Catastrophic.

A.2.1 Roll Control System Requirements Capture

The process began when the systems engineer for the Flight Controls System reviewed an aircraft-level requirement regarding the aircraft's roll performance. This requirement was allocated to the Flight Controls System as one of many inputs in the generation of the proposed system architecture. During the process of developing the system architecture and associated model, the systems engineer captured system requirements through aircraft-level requirement decomposition and various design decisions. These system requirements included the following performance requirement of the roll control function:

SYS_ROLL_REQ-123: While in level-flight cruise, the Roll Control System shall roll the aircraft to an angle of bank of 30 degrees within 3 seconds of the pilot commanding full authority.

This requirement was then allocated to the aileron controller for further decomposition. The proposed architecture and design of the aileron controller was that of a closed-loop feedback on aircraft roll rate between the pilot commanded and detected position with a proportional gain (gain K_p as indicated in [Figure A1](#)) to respond to the observed difference. This resulted in the following proposed system-level requirement for the aileron controller:

AIL_CONT_REQ-123: The aileron roll control shall command the aileron actuator with a proportional gain of X in a closed-loop feedback of aircraft roll rate between pilot command and aileron position sensor.

For the simplicity of the example, these requirements will be the only ones discussed using this analysis model. At this stage in the development process, there was not yet enough knowledge to propose a proportional gain. Initial analysis was required to evaluate the accuracy of the analysis model.

NOTE: The focus of this example is on the use of models to develop and validate system performance requirements; as such, these requirements are intentionally simplified.

A.2.2 Roll Control System Planned Requirements Validation

The systems engineer then planned the requirements validation activities to be performed for the proposed aileron controller requirement (AIL_CONT_REQ-123), specifying the use of simulation and test to complement traceability and engineering review (refer to ARP4754/ED-79 [at latest revision]). Their selection was made for three primary reasons:

- Requirements traceability and engineering review, while necessary to establish requirement completeness, cannot alone fully validate the proposed aileron controller requirement for correctness due to the interaction of the aileron controller function with human operators and flight dynamics.
- Simulation can evaluate the roll control function under a wide range of flight and performance scenarios in an environment representative of aircraft dynamics.
- Test can evaluate the roll control function with the actual aircraft flight dynamics and human operators in the fully integrated product.

While not a primary reason for selection, simulation was also used for the reduction of risk for identifying significant errors in the requirements prior to implementation and test.

NOTE: The use of the term "simulation" corresponds to performing the requirements validation method of "Modeling" from ARP4754/ED-79 (at latest revision).

A.2.3 Roll Control System Model Initial Development

At the time of architecture development, an initial version of the Roll Control System model was produced per the modeling standards to assess the viability of the proposed architecture and design. The Roll Control System model consisted of two elements: the plant model for the integrated aircraft/hardware dynamics (including actuation), and the aileron controller model for the algorithm converting pilot inputs into surface commands.

The Roll Control System model was optimized during its development around the flight envelope for normal operation, given the allocated aircraft-level requirements. As such, this model had a limitation in its ability to analyze the Roll Control System in other flight conditions, such as high angle-of-attack. It is important to note that the development of a full plant model would consider various other flight conditions and functions beyond those discussed in this example. As such, additional potential model error sources specific to other flight phases will not be discussed.

The aileron controller model for analysis was developed to reflect the proposed design architecture as a closed-loop feedback on aircraft roll rate. This feedback would be between the pilot commanded and detected positions with a gain to respond proportionally to the observed difference. The design engineers elected to model the Roll Control System (plant and aileron controller) as a first order linear system, primarily to simplify the plant model for analysis. The linear model simulation results were planned to be correlated with a full-scale non-linear model in order to confirm the accuracy of this linear approximation. The non-linear model had been previously validated against similar flight controls architectures performing a similar function. A user guide was created to ensure the model would be used according to the limitations of the linear model.

Model Validation of the Roll Control System model was planned to be achieved through three activities:

- Desktop analysis – comparison of the simplified linear Roll Control System model with the previously validated non-linear model
- System lab test – comparison of the simplified linear Roll Control System model and the system implementation (hardware and software) with modeled aircraft dynamics
- Aircraft test – comparison of the modeled aircraft dynamics with actual aircraft dynamics for the fully implemented aircraft

A.2.4 Roll Control System Model Desktop Analysis

The proportional gain constraint needed to be selected in order to finalize the aileron controller requirement before allocating it to the software. The aileron controller design engineer initially selected a proportional gain of 2.0 based on similar aileron controllers they had developed on products providing similar functionality.

For this example, the system model was simulated with the initial attempt of linearization of the plant model. However, in the engineering review of the simulation results, the plant model was found to exhibit an unexpected form. Further assessment discovered that an erroneous transfer function was used for one of the linearized model elements. This transfer function element was corrected in the plant model, and subsequent simulation was performed. These simulation results were again peer reviewed, and found to be an acceptable representation of the proposed system architecture and design, given the simplifications.

Once the linearized model was determined to be sufficiently accurate at this stage of development, the model was placed under configuration control and simulations were executed by the aileron controller design engineer to establish control gains for the aileron system that would meet the rate requirement.

[Figure A2](#) shows the comparison of the simplified linearized model results with the full-scale non-linear model results, including the effects of actuator position and rate limiting which are not present in the linear model. Assessment of the differences between the linear and full-scale model characteristics showed that they were within expected bounds. As a result, the linear model was initially validated to support linear analysis. The linearized model was subsequently accepted for use in the aileron controller design and the validation of related requirements.

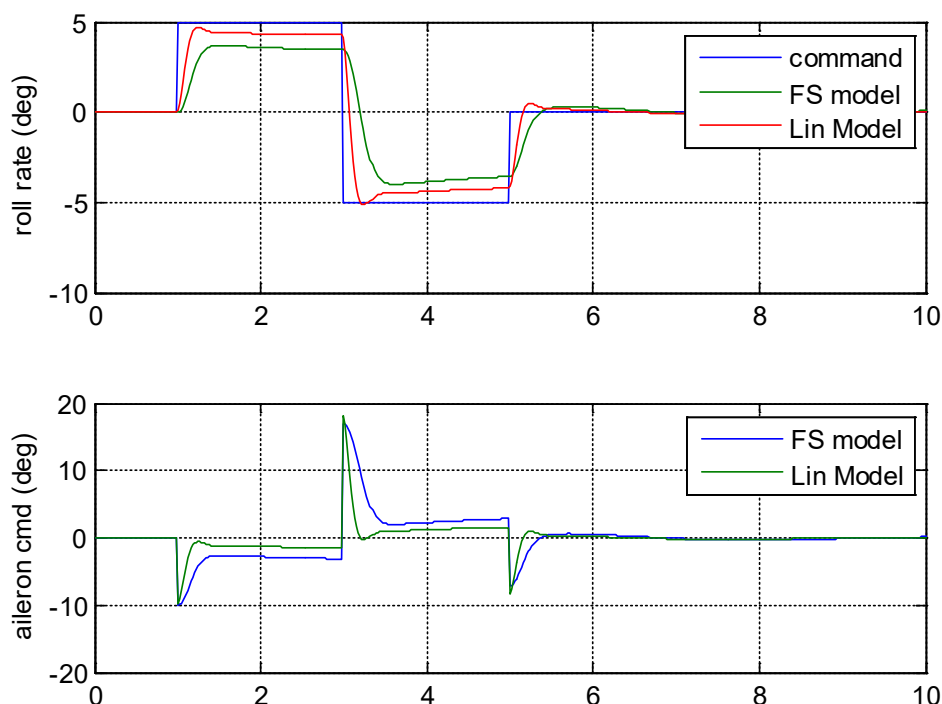


Figure A2 - Linear and non-linear model results for command and roll rate

The system-level requirement defining the behavior of the aileron controller (AIL_CONT_REQ-123) was validated to require converting detected roll rate via a proportional gain of 2.0 to provide the roll command to the actuator. The Validation Matrix for the requirement AIL_CONT_REQ-123 was updated to record evidence of the simulation performed. This requirement was subsequently allocated to the aileron controller software.

AIL_CONT_REQ-123: The aileron roll control shall command the aileron actuator with a proportional gain of 2.0 in a closed-loop feedback of aircraft roll rate between pilot command and aileron position sensor.

A.2.5 Roll Control System Lab Test

While the software was being implemented, the systems engineers prepared for testing to validate the model of the aileron controller and verify the implementation of the aileron controller requirement (AIL_CONT_REQ-123). The planned scenarios identified were developed into a system lab test procedure that was reviewed for its ability to elicit the aircraft roll expected functionality. The system lab test environment planned to execute this test procedure contained real hardware and software for the Roll Control System.

After aileron controller implementation via the software development process was complete, the software was used in this system lab test environment to collect data on the Roll Control System functionality. These test results were used to correlate the prior Roll Control System model simulation results.

The system lab test results did not provide the same response exhibited by the linearized model results and also did not meet the requirement to achieve a 30-degree bank in 3 seconds. Analysis of the system lab test results was completed, and it was determined that the physical implementation contained a feedback time constant that was not represented in the plant model, thereby resulting in a lag in physical system response. This delay was added to the plant model, and a new simulation of the updated model showed an acceptable response compared to the system lab test results. The correlation of the updated Roll Control System model to the system lab test results enabled additional incremental model validation.

The Roll Control System model was then simulated to re-assess the control gain, and it was determined that the aileron controller requirement for a proportional gain of 2.0 was also not the correct specification as it took 4 seconds to achieve the parent requirement roll rate characteristic. Thus, an increased proportional gain of 2.5 was proposed to solve the issue so as to improve the system's roll performance without making it uncontrollable. Subsequently, the previously accepted linearized model was updated and desktop analysis (simulation) re-performed. The new simulation results confirmed that the proposed updated proportional gain demonstrated that the aileron controller behavior was likely to satisfy the parent roll rate requirement. As such, the proportional gain was changed from 2.0 to 2.5 in the aileron controller requirement and subsequently re-flown down to software where the implementation was modified.

AIR_CONT_REQ-123: The aileron roll control shall command the aileron actuator with a proportional gain of 2.5 in a closed-loop feedback of aircraft roll rate between pilot command and aileron position sensor.

Upon receiving the updated system configuration, the system lab tests were performed again; the control response matched the model results and the roll rate of 30 degrees in 3 seconds was achieved. The Validation Matrix for the aileron controller requirement was updated to record evidence of the updated simulation performed.

A.2.6 Roll Control System Aircraft Flight Test

After the control software was released for use on the aircraft, flight test results were evaluated to both:

- Validate the flight dynamics portion of the Roll Control System model
- Assess whether the Roll Control System requirement (SYS_ROLL_REQ-123) was being met by the implemented Roll Control System

The flight test results did show that the roll requirement was being met without the presence of any unintended behavior. The same flight test results were correlated with simulation results of the Roll Control System model utilizing the same test procedure. Correlation validated the Roll Control System model as an acceptable representation of the aircraft flight dynamics.

A.3 ROLL CONTROL SYSTEM MANAGEMENT OF MODEL RELATED RISK

A determination of the risk involved in developing the Roll Control System model included a summary of the contribution of the model to the Roll Control System design, the level of criticality of the model that was being designed, as well as the steps taken to mitigate the potential errors discussed in the main body of the document.

A.3.1 Contribution of Roll Control System Model to System Development

The Roll Control System, and specifically the aileron controller, was designed with the assistance of corresponding plant and aileron controller models. These models were tasked with analyzing the Roll Control System behavior to identify requirements and design issues early in the development process. There were many planned activities to ensure the correctness of the model in representing the aircraft roll dynamics and roll control behavior through the application of desktop analysis (simulation), system lab test, and aircraft flight test. The rigor planned in the development process for the Roll Control System modeling was commensurate with the use of the Roll Control System model in the Roll Control System development.

A.3.2 Risk Assessment of Roll Control System Model Usage

In this example, the Roll Control System was modeled to analyze the proposed flight critical aileron roll control function in support of requirements capture and validation (correctness), and to avoid late and costly discoveries of design issues to address. This functionality was identified through the System Safety Assessment to potentially result in Catastrophic effects if it performs erroneously. This means that improperly selected aileron controller gains could introduce erroneous levels of aileron commands leading to these Catastrophic effects. As such, the criticality of the aileron controller gain selection was a major consideration in how the modeling process was planned to be mitigated through increasing integration of hardware in the loop, up to and including aircraft flight test.

A.3.3 Roll Control System Model Error Mitigation

Incremental plant and aileron controller model development and validation provided a means to progressively reduce risk of the effects of a modeling error escaping to successive stages where it becomes more expensive to address. In the example given, an error was detected in the earliest stages; each successive stage provided an opportunity to detect and mitigate undesired behavior arising from modeling approximations or errors of progressively decreasing likelihood.

Model errors and limitations will become apparent during model validation activities based largely on correlation to the physical objects being modeled. Due to the late stage of that activity, interim means such as engineering assessment based on experience, legacy data, and modeling standards are appropriate.

Ultimately the performance of the integrated aileron controller and plant model was validated against the physical system with human pilots in the loop. This served to additionally mitigate modeling errors and to substantiate use of the model for verification of performance over many more conditions than can be applied to the physical system.

The planned mitigations and their justification for sufficiency were chosen given the overall consequences of the model's usage in the system development process noted by the risk assessment. This included the consideration for minimizing the potential for residual errors after implementing the planned mitigations commensurate with the identified consequences. [Table A1](#) indicates the identified error sources, the selected mitigations, and their justifications. It should be noted that this example focuses on the contribution of the system model and not the tools used in model development or execution. As such, potential error sources from the system tool and other necessary mitigations have been omitted for simplification.

Table A1 - Roll control system model planned mitigations

Error Types	Mitigations Used	Mitigation Description	Justification/Rationale/Limitation
Development Process Errors	Modeling Standards	All engineers involved in the model development process follow the same modeling standards.	Following the modeling standards ensures consistency of the elements and interfaces of the plant and aileron controller model elements.
	Model Validation	A qualitative assessment is completed of the model output (plant and aileron controller combined) and internal state response to simple stimulus, and subsequently the model outputs are correlated to known reference models, system lab test results, and aircraft test results.	Potential errors in the developed Roll Control System model that comply with the Modeling Standard will be found through Model Validation to confirm it acceptably represents the Roll Control function.
Configuration Management Errors	Configuration Management	The model is managed such that each piece (plant and aileron controller) is uniquely identified for usage, only contains approved modifications, and is only available through approved sources. NOTE: Only one version of the model (plant and aileron controller combined) at a time is planned for use throughout this project. Any model revisions will be made using the configuration management framework.	Potential errors in one version of the Roll Control System model that can manifest in incorrect results are mitigated by utilizing configuration control for creating a new version of the Roll Control System model for subsequent analysis.
Representation Errors	Peer Review (Preliminary)	Comparison to previous models of similar aircraft is done as a review of plant model elements and their derivation.	The structure and methodology of construction of plant models for many categories of aircraft have been well established and can be used to review implementation of the physics and timing factors in the plant model. Not applicable for the aileron controller model as it is a new control design.
	Peer Review (Implementation)	A qualitative assessment is completed of the model output (plant and aileron controller combined) and internal state response of the aileron controller model to simple stimulus.	Potential representation errors in the Roll Control System model (plant and aileron controller) representing the aileron control can manifest in the results not matching the expected system response. These incorrect results are mitigated by reviewing the Roll Control System model design and the Roll Control System model results against separate manual analysis results and other, more-detailed, reference models.

Error Types	Mitigations Used	Mitigation Description	Justification/Rationale/Limitation
Representation Errors (cont.)	Model Validation	A qualitative assessment is completed of the model output (plant and aileron controller combined) and internal state response to simple stimulus, and subsequently the model outputs are correlated to known reference models, system lab test results, and aircraft flight test results.	Simplifications in the Roll Control System model representing the aileron control, such as linearization of a non-linear system, can manifest in the results not matching the expected system response. These incorrect results are mitigated by validating the Roll Control System model (plant and aileron controller combined) results against non-linear, known reference models and actual hardware where possible.
	Downstream Implementation Verification	The implementation is verified during system lab test and aircraft flight test to ensure that the aileron controller system-level requirement and the parent system-level requirement are met by the implemented system in the environment.	Verification of the aileron control design implementation on the aircraft further confirms that the plant model acceptably represents the actual flight dynamics. It also validates the assumptions made regarding pilot behavior during the roll event. NOTE: Implementation Verification can identify a number of errors including model representation errors. However, it can't identify the source of error.
Computational Errors	Peer Review (Implementation)	A qualitative assessment is completed of the model output (plant and aileron controller combined) and internal state response of the aileron controller model to simple stimulus.	Potential computation errors and latency effects in the Roll Control System model (plant and aileron controller combined) representing the aileron control can manifest in the results not matching the expected system response. These incorrect results are mitigated by reviewing the Roll Control System model design and the Roll Control System model results for acceptability by experienced design engineers.
	Model Validation	A qualitative assessment is completed of the model output (plant and aileron controller combined) and internal state response to simple stimulus, and subsequently the model outputs are correlated to known reference models, system lab test results, and aircraft flight test results.	Simplifications in the Roll Control System model representing the aileron control, such as linearization of a non-linear system, can manifest in the results not matching the expected system response. These incorrect results are mitigated by validating the Roll Control System model (plant and aileron controller combined) results against non-linear, known reference models and actual hardware where possible.
Errors Resulting from Inapplicable or Improper Usage	User Guide	The user guide for the Roll Control System model specifies that its plant model is a linear approximation of the system and that the results should be interpreted accordingly.	The user of the plant model must understand its design and limitations, which are described in the user guide to ensure that the plant model is appropriate for the intended use.
Setup Errors	User Guide	The user guide for the Roll Control System model is used to ensure that all setup and initial conditions, such as Mach and Angle-of-Attack or assumed pilot behaviors, are within the covered boundary conditions of the model design, and that the inputs and interfaces are defined as required by the model.	The plant model design and validation covers assumed boundary limits, which must be communicated to the user so that the Roll Control System model is not exercised beyond its design, which could introduce errors. The required form of the inputs and interfaces must also be defined to ensure that the inputs to both the plant and aileron controller portions of the Roll Control System model are valid for the intended use.
Results Interpretation Errors	User Guide	The user guide provides direction on how to interpret the results of the model (plant and aileron controller combined), including potential inaccuracies or simplifications that may affect conclusions of the analysis.	The user of the Roll Control System model needs to be aware of the intended use of the Roll Control System model results, so as to not interpret the outputs in an incorrect context.

APPENDIX B - SYSTEM DEVELOPMENT EXAMPLE: AVIONICS NETWORK EVALUATION

B.1 INTRODUCTION

For this example, an Interface Control Document (ICD) was developed for the avionics network on the fictional S18 aircraft. A tool was used that analyzes the data that each system publishes (i.e., generates outputs) and subscribes (i.e., receives inputs) to verify that the required system interfaces were free of error during the system development process (see [Section B.2](#)).

This example also presents the selected mitigation strategies relevant to the tool development and its application in the system development process relevant to its risk assessment (see [Section B.3](#)).

To focus on the tool impact on the system, a simplified portion of the entire avionics network was chosen. For this example, only the Air Data System (ADS) and the Inertial Reference System (IRS) interfaces and related network paths were considered. The basic physical architecture of the network is as shown in [Figure B1](#).

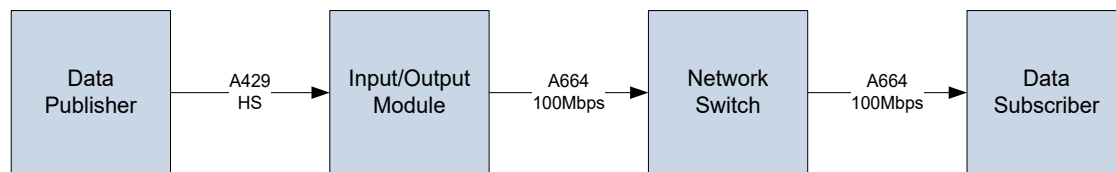


Figure B1 - Avionics network architecture

The Network Analyzer Tool is a computer-based tool that takes the raw interface data as input and provides the following output:

- Report of subscribed data that is being published, of published data that is being subscribed, and of any issues of either published data that is not subscribed or subscribed data that is not being published
- Report of any issues where the publisher is not providing data that matches the specifications of the subscriber
- Report of latency analysis on the end-to-end performance for the data to traverse through the system

B.2 USE OF NETWORK ANALYSIS TOOL IN SYSTEM DESIGN

This section of the example is intended to show how a tool can be used for system analysis, interface requirements assessment, and design risk mitigation. The Network Analysis Tool was used at several points during the process of developing the avionics network. The process began with the systems engineer gathering interface data from each of the item-level providers, which was used as an input to the design of the network. For the purposes of this example, the gathered interface data was an input to the tool and, for simplicity, was not itself considered to be a model.

Once the network had been designed, the interface data was combined into a model of the network which was then run through the Network Analysis Tool in order to uncover any potential issues. The systems engineer confirmed the accuracy of the reported issues and used them along with other considerations, such as the level of customization of each of the interfacing systems (legacy system, modified legacy system, custom system), the criticality of the data parameter, the available bandwidth, and the performance of the producing and/or consuming systems, to make decisions in order to refine the system.

Once the system had been refined, the Network Analysis Tool was again run to confirm that the issues had been resolved. The updated interface data was then reviewed and provided to the item developers in the form of an ICD.

If item developers discover additional issues with the data, further refinements to the network and interface design may be made. These refinements would then be assessed by the Network Analysis Tool, reviewed, and provided back to the item developers.

B.2.1 Tool Selection

This example used a previously developed tool application created by the network design engineers, including its established user guide. Prior to using this tool for performing the network analysis, a peer review assessment was conducted to confirm the suitability of its application. This peer review (concept) confirmed the suitability of the tool with the following observations:

- It appropriately bounds the precision of the latency calculation inputs to limit potential floating point errors
- It appropriately restricts the latency calculation to a simple arithmetic (i.e. summation) methodology rather than more sophisticated strategies
- Eventual Peer Review (Implementation) will be sufficient to identify any incorrect representations of publisher and subscriber data mismatches in the tool results

B.2.2 Tool Input

As a result of inputs from the item developers, analysis of industry interface standards, and the overall system design, the system engineer generated the tool inputs for the interfaces as shown in [Table B1](#).

Table B1 - Interface data initial baseline

Parameter	Interface Type	Bus Name	Speed	Rate (Hz)	Period (ms)	Technical Latency	Publisher	Subscriber	Jitter	Transport Delay [System Latency] (ms)
Pressure Altitude	A429	ADS Data Bus	100 kHz	20	50	Normal	ADS 1-3	IOM	Normal	25
Pitch	A429	IRS Data Bus	100 kHz	40	25	Critical	IRS 1-3	IOM	Critical	12.5
Pitch Rate	A429	IRS Data Bus	100 kHz	80	12.5	Critical	IRS 1-3	IOM	Critical	6.25
Pressure Altitude	A664	I/O Bus	100 Mbps	10	100	Normal	IOM	Network Switch	Normal	50
Pitch	A664	I/O Bus	100 Mbps	80	12.5	Critical	IOM	Network Switch	Critical	6.25
Roll	A664	I/O Bus	100 Mbps	40	25	Critical	IOM	Network Switch	Critical	12.5
Pressure Altitude	A664	Network Bus	100 Mbps	10	100	Normal	Network Switch	Displays 1-5	Normal	50
Pitch	A664	Network Bus	100 Mbps	80	12.5	Critical	Network Switch	Flight Control 1-3	Critical	6.25
Roll	A664	Network Bus	100 Mbps	80	12.5	Critical	Network Switch	Flight Control 1-3	Critical	6.25

This interface data table serves as the input to the Network Analysis Tool.

B.2.3 Tool Output

Following execution of the Network Analysis Tool, the issues listed in [Table B2](#) were reported.

Table B2 - Interface data issues initial

Issue ID	Issue Type	Parameter	Publisher	Subscriber	Issue Data
1	Subscriber with No Publisher	Roll	N/A	Flight Control 1-3	
2	Data Mismatch	Pressure Altitude	ADS 1-3	Displays 1-5	Rate, Period
3	Data Mismatch	Roll	N/A	Flight Control 1-3	Rate, Period
4	Latency Mismatch	Pitch	IRS 1-3	Flight Control 1-3	Calculated Latency = 25 ms

To confirm the accuracy of the tool output, the systems engineer analyzed the parameter data for each of the issues reported to ensure that the issues reported really exist. Any erroneously reported issues were then removed from the list of issues. In accordance with good engineering practices, the issues retained their original issue IDs.

In addition to analyzing the reported issues, the systems engineer performed a manual analysis of a sample (e.g., 10%) of the parameters for confirmation of no unreported issues. This was done for tool validation in order to develop a greater confidence that the tool output is reasonable and accurate. The engineer selected the sample from the entire network model; however, particular emphasis was given to areas with typically high numbers of issues and high criticality of data, such as high speed and high bandwidth busses. If a significant number of unreported issues had been discovered within this analysis, a full analysis of the model would have been performed. Any unreported issues were added to the list of reported issues and given a unique issue ID. Manual analysis was only one of several methods of confirming the accuracy of the tool output.

For this example, it was determined that issues #1, 3, and 4 were legitimate concerns that should be resolved. However, issue # 2 was a case where the data was being sampled at a lower rate by the subscriber than the rate at which the data was being published. There were no bandwidth issues on the bus, so this was therefore dispositioned as not an issue.

Given the small amount of data in this example, a full manual analysis of all network parameters was performed. As part of this analysis, it was determined that the tool did not detect that IRS 1-3 publishes the Pitch Rate, which is not used by any subscriber, as the tool was not designed to check for missing subscribers. This issue was then added to the list of reported issues.

As a result of these analyses, an updated list of issues was produced in [Table B3](#).

Table B3 - Interface data issues final

Issue ID	Issue Type	Parameter	Publisher	Subscriber	Issue Data
1	Subscriber with No Publisher	Roll	N/A	Flight Control 1-3	
2	Non-Issue				
3	Data Mismatch	Roll	N/A	Flight Control 1-3	Rate, Period
4	Latency Mismatch	Pitch	IRS 1-3	Flight Control 1-3	Calculated Latency = 25 ms
5	Publisher with No Subscriber	Pitch Rate	IRS 1-3	N/A	

Since the version of the Network Analysis Tool used in generating the results in [Table B2](#) did not identify the missing data subscribers, the decision was made to update the Network Analysis Tool regarding issue #5 for future analysis of the network. Note that the role of this tool is for an earlier identification of issues that would ultimately be found in test and integration. As such, the tool does not need to identify all issues, but was updated to improve the results of this earlier analysis.

B.2.4 Use of Tool Output for Design Updates

Due to the issues discovered by the tool, updates were subsequently made to the system interface design.

The Flight Control System required roll data, so issue #1 flagged by the tool resulted in a change to the Inertial Reference System by the item developer to publish the required parameter.

The tool identified roll data was transmitted at 40 Hz by the Network Switch but subscribed at 80 Hz by the Flight Control System (issue #3). The system engineer initiated an analysis of the Flight Control System performance which determined that the Flight Control System performance was met with roll data provided at 40 Hz. Therefore, the Flight Control System was modified by the item developers to receive roll at 40 Hz. Additionally, the Inertial Reference System will publish the roll data at 40 Hz.

No system was using pitch rate data, so issue #5 flagged by the tool resulted in a change to the Air Data System by the item developer to not publish that unused parameter. If there was no technical impact to having unused pitch rate data, it may have been left in the design. This is not a recommended practice in general, but there may be circumstances where it may be preferred. It is the role of the system engineer to make this determination.

The tool computed the system latency of pitch data as 25 ms (sum of all three pieces of the latency), but the requirement was 18.75 ms (issue #4). This resulted in a change to the IRS Data Bus by the system engineer to expedite that data to meet the 18.75 ms performance requirement.

After addressing the issues raised by the tool in the design (see highlighted changes in [Table B4](#)), the system engineer re-initiated the tool to perform another analysis. The system engineer then manually reviewed the output to verify each of the issues that were previously raised were sufficiently addressed and no new undesired behaviors were introduced. The updated tool was then placed under configuration control for future use.

Table B4 - Interface data interim baseline #2

Parameters	Interface Type	Bus Name	Speed	Rate (Hz)	Period (ms)	Technical Latency	Publisher	Subscriber	Jitter	Transport Delay [System Latency] (ms)
Pressure Altitude	A429	ADS Data Bus	100 kHz	20	50	Normal	ADS 1-3	IOM	Normal	25
Pitch	A429	IRS Data Bus	100 kHz	80	12.5	Critical	IRS 1-3	IOM	Critical	6.25
Roll	A429	IRS Data Bus	100 kHz	40	25	Critical	IRS 1-3	IOM	Critical	12.5
Pressure Altitude	A664	I/O Bus	100 Mbps	10	100	Normal	IOM	Network Switch	Normal	50
Pitch	A664	I/O Bus	100 Mbps	80	12.5	Critical	IOM	Network Switch	Critical	6.25
Roll	A664	I/O Bus	100 Mbps	40	25	Critical	IOM	Network Switch	Critical	12.5
Pressure Altitude	A664	Network Bus	100 Mbps	10	100	Normal	Network Switch	Displays 1-5	Normal	50
Pitch	A664	Network Bus	100 Mbps	80	12.5	Critical	Network Switch	Flight Control 1-3	Critical	6.25
Roll	A664	Network Bus	100 Mbps	40	25	Critical	Network Switch	Flight Control 1-3	Critical	12.5

B.2.5 Interface between Avionics Network Development and Item Development Processes

Once the interface data was updated, the system engineer held a data review. Various stakeholders were invited to participate, including additional system engineers, the systems safety group, quality assurance engineers, and each of the item developers. As part of this review, the item developers were given the opportunity to confirm that their items will support the defined interface data.

This review resulted in the item developers determining that another parameter for Barometric Altitude was needed to be generated by the Air Data System and subscribed by the Flight Control System. The item developers changed their designs accordingly and the system engineer updated the interface data (see highlighted changes in [Table B5](#)). The tool was re-run and the system engineer and item developers collectively reviewed the revised output to confirm that the system behaves as intended. Once the review was held, the action items resolved, and the review closed, the interface data was delivered to each of the item developers for implementation.

As part of the program schedule, the item developers were required to implement the interfaces early in their development cycles, with early limited functionality releases, in order to facilitate early system integration. During the system integration activities of the program, informal system verification was performed on the avionics network. During verification, other issues were found regarding the avionics network design unrelated to the tool and its usage in the design process. For example, the system developers of the Flight Control System determined that 10 Hz was not sufficient for the Pressure and Barometric Altitude and requested a change to 20 Hz. The system developers for the Flight Control and Air Data Systems modified their designs accordingly (see highlighted changes in [Table B5](#)), the system engineer updated the interface data and re-ran the tool. The item developers and the system engineer collectively analyzed the output of the tool to verify that the changes were implemented correctly and no undesired behaviors were introduced.

Table B5 - Interface data interim baseline #3

Parameters	Interface Type	Bus Name	Speed	Rate (Hz)	Period (ms)	Technical Latency	Publisher	Subscriber	Jitter	Transport Delay [System Latency] (ms)
Pressure Altitude	A429	ADS Data Bus	100 kHz	20	50	Normal	ADS 1-3	IOM	Normal	25
Barometric Altitude	A429	ADS Data Bus	100 kHz	20	50	Normal	ADS 1-3	IOM	Normal	25
Pitch	A429	IRS Data Bus	100 kHz	80	12.5	Critical	IRS 1-3	IOM	Critical	6.25
Roll	A429	IRS Data Bus	100 kHz	40d	25	Critical	IRS 1-3	IOM	Critical	12.5
Pressure Altitude	A664	I/O Bus	100 Mbps	20	50	Normal	IOM	Network Switch	Normal	50
Barometric Altitude	A664	I/O Bus	100 Mbps	20	50	Normal	IOM	Network Switch	Normal	50
Pitch	A664	I/O Bus	100 Mbps	80	12.5	Critical	IOM	Network Switch	Critical	6.25
Roll	A664	I/O Bus	100 Mbps	40	25	Critical	IOM	Network Switch	Critical	12.5
Pressure Altitude	A664	Network Bus	100 Mbps	20	50	Normal	Network Switch	Displays 1-5	Normal	50
Barometric Altitude	A664	Network Bus	100 Mbps	20	50	Normal	Network Switch	Flight Control 1-3	Normal	50
Pitch	A664	Network Bus	100 Mbps	80	12.5	Critical	Network Switch	Flight Control 1-3	Critical	6.25
Roll	A664	Network Bus	100 Mbps	40	25	Critical	Network Switch	Flight Control 1-3	Critical	12.5

B.3 MANAGEMENT OF TOOL RELATED RISK

A determination of the risk involved in using the Network Analysis Tool included a discussion of the contribution of the tool to the network design, the level of criticality of the network that was being designed, and the confidence that can be gained in using this particular tool, as well as the steps taken to mitigate the potential errors discussed in the main body of the document.

B.3.1 Contribution of the Network Analysis Tool to Development Process

The avionics network was designed prior to the use of the Network Analysis Tool. The purpose of using the Network Analysis Tool was to increase the maturity of the design prior to implementation, thus reducing schedule and cost through identifying design issues at an earlier stage in the development process. The development process for the avionics network met the guidelines of ARP4754/ED-79 (at latest revision), including the validation of interface requirements and integration testing, without using the Network Analysis Tool. Functional and design issues that adversely affect system performance or safety would be detected prior to system deployment. As a result, the contribution of the Network Analysis Tool to the safety of the avionics network was minimal.

B.3.2 Risk Assessment of the Network Analysis Tool Usage

The risk assessment of the Network Analysis Tool identified that it was used to perform early validation of the avionics network design to mitigate late discoveries of design issues that would be very expensive to fix. Implementation verification was planned per ARP4754/ED-79 (at latest revision) guidelines without using the Network Analysis Tool. Thus, the Network Analysis Tool can only have an economic impact on the project with no impact on the safety of the product.

B.3.3 Error Mitigation

During the project planning phase, the Network Analysis Tool and its proposed usage in the system development process were reviewed to identify potential error sources. In this example, the Network Analysis Tool was used to perform early validation of the definition of flight critical data (attitude and altitude) across the avionics network.

The planned mitigations and their justifications for sufficiency were chosen given the overall consequences of the tool's usage in the system development process noted by the risk assessment. This includes the consideration for minimizing the potential for residual errors after implementing the planned mitigations, commensurate with the identified consequences. [Table B6](#) indicates the identified error sources, the selected mitigations, and their justifications.

Table B6 - Network analysis tool planned mitigations

Error Types	Mitigations Used	Mitigation Description	Justification/Rationale/Limitation
Development Process Errors	Tool Validation	Systems engineer will analyze the parameter data for each of the issues reported by the tool, throughout the development process, to ensure that the issues reported really exist. Any incorrectly reported issues will be removed from the list of issues with justification. In addition to analyzing the reported issues, the systems engineer will perform a manual analysis of a representative sample of the parameters to increase confidence that all issues were reported. This will be performed once for the initial version of the tool and will be considered for subsequent versions.	Potential errors in the tool development process that can manifest in the tool producing incorrect results are mitigated by validating the tool initial results against separate manual analysis results. Additional manual analysis for tool validation will cover each type of check that is performed by the tool.
Configuration Management Errors	Configuration Management	The tool is managed such that it is uniquely identified for usage, only contains approved modifications, and is only available through approved sources. NOTE: Only one version of the Network Analysis Tool is planned for use at a time throughout this project. Any tool revisions will be made using the configuration management framework.	Potential errors in one version of the tool that can manifest in the tool producing incorrect results are mitigated by utilizing configuration control for creating a new version of the tool for subsequent analysis.
Representation Errors	Peer Review (Concept)	A review is conducted during the tool development to determine a suitable calculation strategy for the planned latency analysis.	Early peer review ensures the correct representation of latency calculation for the avionics network under analysis. The Network Analysis Tool latency estimation is planned to be a linear calculation (i.e., sum), and the peer review will confirm the results will be of a suitable tolerance for analysis results.
	Peer Review (Implementation)	A review is conducted during the tool execution phase to perform a final check on the suitability of the publisher subscriber data correctness checks that are implemented in the tool.	Peer Review of algorithms implemented in the tool can identify any potential representation errors to detect publisher subscriber data mismatches.

Error Types	Mitigations Used	Mitigation Description	Justification/Rationale/Limitation
	Tool Validation	Systems engineer will analyze the parameter data for each of the issues reported throughout the development process to ensure that the issues reported really exist. Any incorrectly reported issues will be removed from the list of issues with justification. In addition to analyzing the reported issues, the systems engineer will perform a manual analysis of a representative sample of the parameters to increase confidence that all issues were reported. This will be performed once for the initial version of the tool and will be considered for subsequent versions.	Potential errors in the tool results representing the avionics network can manifest in the tool producing incorrect latency calculation results. These incorrect results are mitigated by validating the tool results against separate manual analysis results. Additional manual analysis for tool validation will cover each type of check that is performed by the tool.
	Downstream Implementation Verification	Formal verification of network implementation will be performed per ARP4754/ED-79 (at latest revision) guidelines.	The Network Analysis Tool is used to minimize discovery of design issues during the verification phase. Implementation verification is planned per ARP4754/ED-79 (at latest revision) guidelines with no credit from the tool usage. Implementation Verification can catch a number of errors, including tool representation errors. However, it can't identify the source of error.
Computational Errors	Peer Review (Concept)	A review is conducted during the tool development to determine a suitable calculation strategy for the planned latency analysis.	Early peer review ensures the correctness of the latency calculation for the avionics network under analysis. The Network Analysis Tool latency estimation is planned to be a linear calculation (i.e., sum), and the peer review will confirm the results will be of a suitable tolerance for analysis results.
Computational Errors (cont.)	Tool Validation	Systems engineer will analyze the parameter data for each of the issues reported throughout the development process to ensure that the issues reported really exist. Any incorrectly reported issues will be removed from the list of issues with justification. In addition to analyzing the reported issues, the systems engineer will perform a manual analysis of a representative sample of the parameters for validation with no reported issues. This is being done to develop a greater confidence that the tool output is reasonable and accurate. This will be performed once for the initial version of the tool and will be considered for subsequent versions.	Potential errors in the tool results representing the avionics network can manifest in the tool producing incorrect latency calculation results. These incorrect results are mitigated by validating the tool results against separate manual analysis results. Additional manual analysis for tool validation will cover each type of check that is performed by the tool.

Error Types	Mitigations Used	Mitigation Description	Justification/Rationale/Limitation
Errors Resulting from Inapplicable or Improper Usage	User Guide	The Network Analysis Tool User Guide documents the required tool inputs, how the results will be calculated, and how the tool output will be generated. It will also describe the intended application of the tool, as well as its limitations.	The Network Analysis Tool user guide is available to all team members performing analysis and reduces the likelihood of misinterpreting the tool usage and its results.
Setup Errors	User Guide	The Network Analysis Tool User Guide documents the required tool inputs (including units and initial conditions). It will also describe the tool limitations.	The Network Analysis Tool user guide is available to all team members performing analysis and reduces the likelihood of providing improper tool inputs or initial conditions.
Results Interpretation Errors	User Guide	The Network Analysis Tool User Guide documents how the results will be calculated, and how the tool output will be generated.	For latency results interpretation error, the Network Analysis Tool user guide clarifies the correct use of system boundaries, and the simplifications made in its calculations. Interpretation error for evaluation of data mismatches or data with no publisher or subscriber is not applicable since the results are a binary pass/fail. The Network Analysis Tool user guide is available to all team members performing analysis and reduces the likelihood of misinterpreting the tool results.